

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (Presently Amended) A method for managing a data caching service from a
2 management terminal in a distributed computer system having a host computer
3 system with a host memory and at least one storage device connected to the
4 computer system by driver software, the method comprising:
5 (a) inserting an interface layer between the driver software and the storage
6 device, the interface layer exporting a platform dependent API comprising
7 a plurality of API methods that can be called to control and controlling data
8 passing between the driver software and the storage device;
9 (b) running, in the host computer system, management facade software that
10 converts the interface layer API to receives calls to platform-independent
11 method calls methods and generates at least one API method call to the
12 interface layer in order to execute the platform-independent method calls;
13 (c) running, in the host computer system, a federated bean that generates
14 platform-independent method calls to the management facade software to
15 control the interface layer via the plurality of API methods; and
16 (d) controlling the federated bean to enable a data read cache by instructing
17 the interface layer to intercept requests for data from the storage device
18 and, if the data is in the host memory, to retrieve the data from the host
19 memory, and to retrieve the data from the storage device if the data is not
20 in the host memory.
- 1 2. (Original) The method of claim 1 wherein step (d) comprises controlling the
2 federated bean with a command line interface.

- 1 3. (Original) The method of claim 1 wherein step (d) comprises controlling the
2 federated bean with a graphical user interface.
- 1 4. (Original) The method of claim 1 further comprising:
2 (e) using the federated bean to disable the read cache.
- 1 5. (Original) The method of claim 1 further comprising:
2 (f) using the federated bean to set characteristics of the read cache.
- 1 6. (Original) The method of claim 1 wherein the computer system further comprises
2 non-volatile memory and the method further comprises:
3 (g) controlling the federated bean to enable a data write cache by instructing
4 the interface layer to intercept requests to write data to the storage device
5 and to write the data to the non-volatile memory.
- 1 7. (Original) The method of claim 6 further comprising:
2 (h) enabling a flusher thread to periodically copy data in the non-volatile
3 memory to the storage device.
- 1 8. (Original) The method of claim 7 wherein step (h) comprises enabling a plurality
2 of flusher threads wherein the number of enabled flusher threads is determined
3 by the federated bean.
- 1 9. (Original) The method of claim 6 further comprising:
2 (i) using the federated bean to disable the write cache.
- 1 10. (Original) The method of claim 6 further comprising:
2 (j) using the federated bean to set characteristics of the write cache.

1 11. (Presently Amended) Apparatus for managing a data caching service from a
2 management terminal in a distributed computer system having a host computer
3 system with a host memory and at least one storage device connected to the
4 computer system by driver software, the apparatus comprising:
5 an interface layer inserted between the driver software and the storage
6 device, the interface layer exporting a platform dependent API comprising a
7 plurality of API methods that can be called to control ~~and controlling~~ data passing
8 between the driver software and the storage device;
9 management facade software that runs in the host computer system and
10 ~~converts the interface layer API to~~ receives calls to platform-independent method
11 ~~calls~~ methods and generates at least one API method call to the interface layer in
12 order to execute the platform-independent method calls;
13 a federated bean that runs in the host computer system and generates
14 platform-independent method calls to the management facade software to control
15 the interface layer via the plurality of API methods; and
16 a presentation program that controls the federated bean to enable a data
17 read cache by instructing the interface layer to intercept requests for data from
18 the storage device and, if the data is in the host memory, to retrieve the data
19 from the host memory, and to retrieve the data from the storage device if the data
20 is not in the host memory.

1 12. (Original) The apparatus of claim 11 wherein the presentation program
2 comprises a command line interface.

1 13. (Original) The apparatus of claim 11 wherein the presentation program
2 comprises a graphical user interface.

1 14. (Original) The apparatus of claim 11 wherein the presentation program controls
2 the federated bean to disable the read cache.

- 1 15. (Original) The apparatus of claim 11 wherein the presentation program controls
2 the federated bean to set characteristics of the read cache.
- 1 16. (Original) The apparatus of claim 11 wherein the computer system further
2 comprises non-volatile memory and wherein the federated bean comprises
3 methods that enable a data write cache by instructing the interface layer to
4 intercept requests to write data to the storage device and to write the data to the
5 non-volatile memory.
- 1 17. (Original) The apparatus of claim 16 further comprising a flusher thread that
2 periodically copies data in the non-volatile memory to the storage device.
- 1 18. (Original) The apparatus of claim 17 further comprising a plurality of flusher
2 threads wherein the number of enabled flusher threads is determined by the
3 federated bean.
- 1 19. (Original) The apparatus of claim 16 wherein the federated bean comprises
2 methods that disable the write cache.
- 1 20. (Original) The apparatus of claim 16 wherein the federated bean comprises
2 methods that set characteristics of the write cache.
- 1 21. (Presently Amended) A computer program product for managing a data caching
2 service from a management terminal in a distributed computer system having a
3 host computer system with a host memory and at least one storage device
4 connected to the computer system by driver software, the computer program
5 product comprising a computer usable medium having computer readable
6 program code thereon, including:

an interface layer inserted between the driver software and the storage device, the interface layer exporting a platform dependent API comprising a plurality of API methods that can be called in order to control and controlling data passing between the driver software and the storage device;

management facade software that runs in the host computer system and ~~converts the interface layer API to~~ receives calls to platform-independent method ~~calls~~ methods and generates at least one API method call to the interface layer in order to execute the platform-independent method calls;

federated bean software that runs in the host computer system and generates platform-independent method calls to the management facade software to control the interface layer via the plurality of API methods; and

a presentation program that controls the federated bean to enable a data read cache by instructing the interface layer to intercept requests for data from the storage device and, if the data is in the host memory, to retrieve the data from the host memory, and to retrieve the data from the storage device if the data is not in the host memory.

22. (Original) The computer program product of claim 21 wherein the computer system further comprises non-volatile memory and wherein the federated bean comprises methods that enable a data write cache by instructing the interface layer to intercept requests to write data to the storage device and to write the data to the non-volatile memory.

23. (Original) The computer program product of claim 22 further comprising program code that creates a flusher thread that periodically copies data in the non-volatile memory to the storage device.

24. (Original) The computer program product of claim 23 further comprising program code for creating a plurality of flusher threads wherein the number of enabled flusher threads is determined by the federated bean.

1 25. A computer data signal embodied in a carrier wave for managing a data caching
2 service from a management terminal in a distributed computer system having a
3 host computer system with a host memory and at least one storage device
4 connected to the computer system by driver software, the computer program
5 data signal comprising:
6 program code for creating an interface layer inserted between the driver
7 software and the storage device, the interface layer exporting a platform
8 dependent API comprising a plurality of API methods that can be called to control
9 ~~and controlling~~ data passing between the driver software and the storage device;
10 management facade software that runs in the host computer system and
11 ~~converts the interface layer API to~~ receives calls to platform-independent method
12 calls methods and generates at least one API method call to the interface layer in
13 order to execute the platform-independent method calls;
14 federated bean software that runs in the host computer system and
15 generates method calls to the management facade software to control the
16 interface layer via the plurality of API methods; and
17 a presentation program that controls the federated bean to enable a data
18 read cache by instructing the interface layer to intercept requests for data from
19 the storage device and, if the data is in the host memory, to retrieve the data
20 from the host memory, and to retrieve the data from the storage device if the data
21 is not in the host memory.